

I. REMARKS/ARGUMENTS

Claim Status

Claims 1-31 are pending. Claims 1-15 and 18-31 stand rejected. Claims 16 and 17 stand objected to as allowable if rewritten in independent form. Applicant appreciates the examiner's indication of allowable subject matter. Applicant maintains the patentability of claims 1-15 and 18-31 and respectfully requests reconsideration and withdrawal of the rejections and objections to claims 1-31.

Claim Rejections - 35 U.S.C. § 103(a)

Claims 1-15 and 18-31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Horwat (U.S. Patent No. 6,021,275) (hereinafter "Horwat") in view of Chan et al. (U.S. Patent No. 5,339,419) (hereinafter "Chan").

The invention takes a novel approach to emulation that overcomes the shortcomings of conventional emulation. According to the invention, the target program to be emulated is first statically translated to a series of instructions of an intermediate instruction set. The *intermediate instruction set is an instruction set that is optimized for interpretation on the host computer*, rather than being the native instruction set of either the target program or the host computer. The series of instructions is then executed by interpretation on the host computer. Because, the intermediate instruction set is an instruction set that is optimized for interpretation on the host computer, the execution of the series of instructions by interpretation is generally faster than conventional interpretation.

The invention, as recited in claims 1, 9, 19, and 27, includes features that are not disclosed or suggested by the cited references either taken alone or in combination. The features are represented by claim 1:

1. A method for emulating the execution of a target program comprising instructions of an instruction set of a target on a host computer having a different instruction set, said method comprising:
 - performing a static translation of the instructions of the target program into a series of instructions of an intermediate instruction set, *the intermediate instruction set being optimized for interpretation on the host computer*; and
 - executing the series of instructions of the intermediate instruction set *by* interpretation on the host computer. (emphasis added)

Horwat is directed to translating program code so that it may be executed on another machine having a different architecture. More particularly, Horwat focuses on only one aspect of such translation, namely endian-format translation that provides endian-independent representations of data, pointer data, operands, and pointer operands. That is, some machines may be big-endian and other machines may be little-endian. Big-endian format maps the lowest address to the highest order data byte and little-endian format maps the lowest address to the lowest order data byte. When translating between machines having different endian formats, bytes of each instruction or each piece of data may be reordered for the architecture of a second machine. Horwat discloses such reordering of bytes.

Horwat does not disclose or suggest the invention as claimed. First, Horwat only discloses *translation* as the second step in running a program (Horwat Figure 2). That is, Horwat does not disclose *executing instructions by interpretation*. On the contrary, Horwat notes that Intercode object code is provided and therefore the object code can be run with an Intercode *translator* (Horwat col. 8, lines 4-14). Translation, however, is very different from interpretation, as described in more detail above and as acknowledged in the background of Horwat (Horwat col. 1, lines 22-49). Therefore, Horwat does not disclose *executing instructions by interpretation*, as recited by the claims.

The examiner cites to Horwat at col. 23, lines 18-20 for the proposition that Horwat discloses executing instructions by interpretation. In this section, Horwat merely uses the word “interpreted,” but this in no way discloses executing instructions by interpretation. In fact, Horwat uses the term “interpreted” in a completely different context. That is, Horwat describes that during relocation of bytes, a base field of a relocation term may be used to describe how an argument of the relocation term should be interpreted. In this manner, Horwat provides endian-independent representations of data. Horwat, however, does not ever disclose execution of instructions by interpretation.

Nor does Horwat disclose *an intermediate instruction set that is optimized for interpretation*; rather, Horwat merely discloses that during translation, performance-critical sections of a program can be *optimized for a particular target architecture* (Horwat col. 7, lines 7-12). Disclosing that sections of a program can be optimized during translation is not the same as statically translating a target program into instructions of an *intermediate*

instruction set that is optimized for interpretation. Therefore, Horwat does not disclose *an intermediate instruction set that is optimized for interpretation*, as recited by the claims.

The examiner cites to Horwat at col. 7, lines 3-13 for the proposition that Horwat discloses an intermediate instruction set being optimized for interpretation on a host computer. In this section, however, Horwat simply states that a *program* can be optimized for a target architecture. In contrast, the invention claims an *instruction set* that is optimized for interpretation on a host computer. That is, an optimization is not necessary because the *instruction set itself* is already optimized for interpretation on the host computer (although further optimization could be included).

Another difference is that Horwat discloses generating an object code format that is *translated at run time* (Horwat column 3, lines 2-5) and therefore discloses *dynamic translation*. As such, Horwat does not disclose *static translation*, as recited by the claims. As can be appreciated, Horwat does not disclose or suggest *many* of the features recited by the claims.

Chan does not cure the *multiple* deficiencies of Horwat. Chan is directed to a compiler intermediate language that allows a program to be written in machine independent format and also allows the program to be distributed to different types of computers. Chan discloses a low level optimizer that performs optimizations on a low-level compiler intermediate representation (Chan at col. 60, lines 35-45). The examiner cites to Chan for the proposition that it discloses “translating statically optimizing into intermediate instructions.” (emphasis added). Applicant notes, however, that “translating statically optimizing into intermediate instructions” is not the language of the claims and the examiner seems to be reading an optimization step into the claims. The claims, in contrast to the examiner’s apparent interpretation, recite “an *intermediate instruction set*” that is optimized for interpretation on a host computer. It is the *instruction set itself* that is optimized for interpretation on a host computer. Chan’s step of optimizing low-level code is completely irrelevant because the claims recite an instruction set that is optimized for interpretation on a host computer, not a step of optimizing code (although further optimization could be included in the invention).

Using a baseball analogy to further illustrate, let’s assume that there are two types of baseball gloves available: a one-size-fits-all glove and a custom-made glove. The one-size-

fits-all glove is analogous to a conventional instruction set and the custom-made glove is analogous to an optimized instruction set. Any fielder can use the one-size-fits-all glove, but that fielder may not perform very efficiently if the fielder's hand doesn't happen to be a good match for the one-size-fits-all glove. On the other hand, the custom-made glove, because it has been made to perfectly fit the fielder's hand, should result in better fielder performance. The fielder does not have to do anything in particular each time they use the custom-made glove (*i.e.*, no optimization step is required as implied by the examiner). The custom-made glove should simply work better because it fits better. (The fielder could also choose to perform some optimization each time the glove is used, such as oiling the glove, tightening the strings, etc.)

Thus, neither reference, either alone or in combination, teaches or suggests performing static translation of a program on one machine into a series of instructions of an intermediate instruction set, followed by interpretation of the series of instructions of that intermediate instruction set on a second machine, as recited in independent claims 1, 9, 19, and 27. Inasmuch as claims 2-8, 10-18, 20-26, and 28-31 depend from one of claims 1, 9, 19, and 27, these claims are patentable at least by their dependency. For the foregoing reasons, applicants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 1-15 and 18-31 and the objections to claims 16 and 17.

Conclusion

For the foregoing reasons, applicants respectfully submit that all of the claims of the present application patentably define over the cited references of record, alone or in combination. Reconsideration of the office action and an early notice of allowance are respectfully requested. In the event that the examiner cannot allow the present application for any reason, the examiner is encouraged to contact the undersigned attorney, Raymond N. Scott Jr. at (215) 564-8951, to discuss resolution of any remaining issues.

DOCKET NO.: TN137/USYS-0111
Application No.: 09/294,617
Office Action Dated: February 28, 2003

PATENT

Respectfully submitted,



Raymond N. Scott, Jr.
Attorney for Applicant
Registration No. 48,666

Date: May 28, 2003

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439